A Framework for Link Sharing in Cooperative Cross-Media Information Spaces

Beat Signer[1] and Alexandre de Spindler[2] and Moira C. Norrie[2]

[1] Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussels, Belgium
bsigner@vub.ac.be
[2] Institute for Information Systems, ETH Zurich
CH-8092 Zurich, Switzerland
{despindler,norrie}@inf.ethz.ch

Abstract. We present a peer-to-peer version of a cross-media link server that allows users to create and share links between arbitrary forms of digital and physical media. As a specific example, we describe how it could be used to support collaborative forms of annotation of paper and/or digital design documents.

1 Introduction

The concept underlying hypermedia systems of being able to freely link arbitrary resources together has been being adopted to span digital and physical spaces. A fully generalised cross-media link server enables the model of information publishing and access on the web to be extended to any type of information resource and service, and to possibly link these to physical objects.

We have developed such a cross-media link server, called iServer, and used it to develop a wide-variety of applications ranging from mobile tourist information systems to interactive media art installations. In most of these applications, users browse pre-authored information and access services based on the defined information space. However, we also wanted to support cooperative, community-based models of cross-media information spaces, where users can publish, not only their own resources, but also links that they have created between existing resources as proposed in open hypermedia systems. We have therefore developed a peer-to-peer (P2P) version of iServer that enables users to share their link spaces. In this paper, we describe how it could be used to support the sharing of information in a meeting scenario.

We start in Sect. 2 with a motivation for cooperative cross-media information spaces and a discussion of related work. The iServer framework and its extension for cooperative information spaces is presented in Sect. 3. We then present our P2P architecture in Sect. 4 and describe an application of the framework in Sect. 5. After a discussion of the presented approach in Sect. 6 some concluding remarks are given in Sect. 7.

2 Motivation

Nowadays digital information is often not only managed by a single application but associated with other digital resources forming part of a general information space. Hypertext and hypermedia models consider such an information space to be a connected graph where the nodes are resources and the edges are links. The source and target of a link can be either an entire resource or an element within a resource. In addition to the handling of associations between digital resources, physical hypermedia systems enable real-world objects to be linked to digital media, and vice versa, by allowing physical resources to also be included as nodes in the connected information space [2].

For improved flexibility, links can be managed in separate linkbases rather than embedding them in resources [5]. One of the benefits of this approach is that new links can easily be created by the consumers as well as the publishers of resources. Specifically, it allows annotations to be created by users where, not only the link, but also the target content created by the user, can be managed by a link server [3]. Such an open authoring system allows the information space to evolve over time based on user-generated links and content.

For example, participants of a design meeting could easily add their own links and annotations to a shared design space as shown in Fig. 1. In such a meeting, each participant might bring their own laptop or other mobile devices containing their personal resources as well as link information. During a meeting, users will not only work on their laptop but often also with dedicated presentation and interaction devices. These interaction devices may include large interactive surfaces such as interactive wall projections or interactive table surfaces. In addition to the digital devices for information navigation and capture, in many design meetings, paper documents still play an important role in supporting discussions as well as the creation of new, or annotation of existing, information.

If users can create their own links between resources during a design meeting, it is also desirable that this information can be shared with other participants to establish a community-based information space. Since there should still be a certain level of control about the sharing of information, a user has to be able to define if a new link is *private*, *shared within a specific group* (e.g. participants of the meeting) or *public* and accessible to everyone.

Instead of using a central server, a P2P architecture provides the potential for more dynamic forms of sharing such as in the described design meeting scenario where teams of people come together and information can be shared directly among personal mobile devices including laptops and fixed interactive surfaces. As explained later, also annotations captured from working with interactive paper documents could be linked into the cooperative information space.

An important detail about the presented scenario for sharing link information is that the supplemental shared information is not automatically stored persistently on each user's personal device. The shared information rather is treated as a supplemental transient annotation layer on top of the user's personal information space and is stored persistently only on an explicit user request. Furthermore, we do not want to make any assumptions about the resources available on

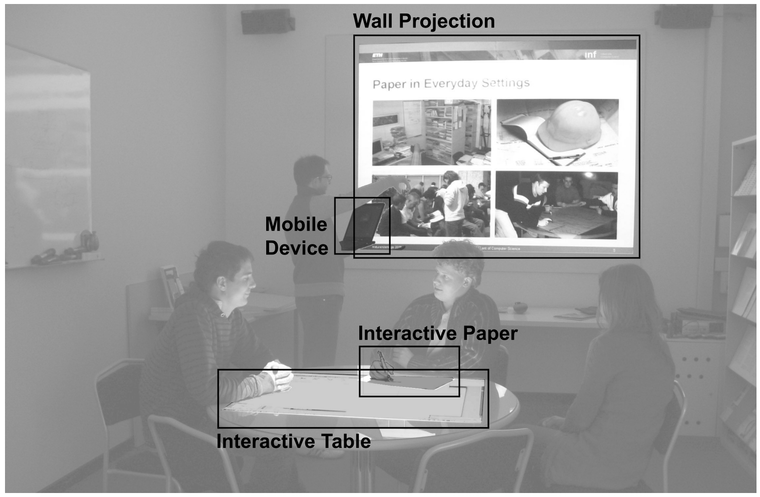

Fig. 1. Cooperative information space

an individual user's device. For example, if some participants of a meeting are from the same company, they tend to have similar resources which on the other hand might not be available to participants from other companies. In this case, the shared links could be filtered to prevent participants from receiving links between resources they do not have on their machine. The sharing of information is therefore not only based on the vicinity of different users but also depends on the availability of common resources.

While the idea of community link sharing behind our framework is similar to proposals in distributed hypermedia [1], the difference lies mainly in the approach to realising the goal. By building on advanced database technologies and concepts of metamodel-driven system engineering, we were able to develop a system that supports all of the above goals within a relatively short period of time. In the rest of this paper, we describe the architecture and implementation of the framework.

3 Cooperative iServer

The iServer framework offers a general cross-media link service capable of supporting an extensible set of digital or physical media and various kinds of applications. The framework is based on the RSL metamodel [8] and enables the integration of new media types based on a resource plug-in mechanism as indicated in Fig. 2.

In the RSL model, links are associated with one or more source entities and one or more target entities. An entity may be a resource, a selector or a link. While a resource represents an entire information unit or service, a selector is used to address parts of a resource, similar to the reference objects introduced in the FOHM [6] model. The RSL model and its iServer implementation further provide some user management functionality that allows access rights to be

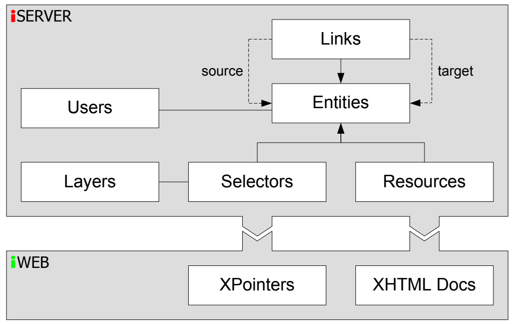

Fig. 2. iServer with iWeb plug-in

associated with single entities (and therefore also with links). Users may be either individuals or groups and each entity has exactly one creator who defines its access rights.

For a particular media type, we can extend the iServer platform by introducing a component that defines specific selectors and resources for that media type. For example, the iWeb plug-in for XHTML documents uses XPointer expressions as selectors to address parts of an XHTML document. In addition to the iWeb resource plug-in, we have developed iServer plug-ins for images, sounds and movies, as well as an RFID plug-in for tagging physical objects. Furthermore, the iPaper plug-in allows regions within paper documents to be linked to digital media and services based on Anoto's digital pen and paper technology[3].

To support the type of enhanced ad-hoc information sharing motivated in the previous section, we developed a P2P framework for iServer. In combination with the iPaper plug-in, this allows users to easily create and share handwritten annotations of paper documents in addition to any other form of link between resources or digital annotations. As mentioned earlier, we do not share the resources themselves but only new associations defined between existing resources. However, a handwritten annotation is a special type of link where the link target is represented by the newly created note. For these special types of annotation links, we therefore not only share the link information but also the captured annotation which is the target resource.

A variety of applications have been implemented using the iServer framework, including general browsers for cross-media information spaces, a paper-based mobile tourist information system [7] and an art installation for writing and accessing interactive cross-media narratives [9].

Existing proposals for cooperative hypermedia systems based on P2P technology, e.g. [4], apply P2P network functionality to replicate hypermedia documents and perform searches in the distributed network of peers. In contrast, we manage the resources separately from the annotations and links and use the P2P architecture only to share link and annotation data.

[3] http://www.anoto.com

In our approach, a client always accesses a resource from its original location and, only in a second step, is additional link metadata acquired over the dynamic P2P iServer network. The separation of content and metadata further implies that a resource should always be available provided that its host web server is running, whereas additional link information may change dynamically based on the set of iServer peers currently available in the network.

A user can access information from their personal iServer link database as well as retrieving link data from any other iServer peer. We distinguish between *persistent link data* that is stored in a personal iServer instance and *transient link data* received from the set of remote iServer peers. The availability of additional link information over the P2P network can be seen as an optional extension to information stored in a personal iServer instance. Users are free to use the new cooperative iServer functionality or to work solely with their personal iServer instance. The transient link data provided by other iServer peers represents optional suggestions by the members of the collaborative information space. However, if a user finds any suggested link information relevant, they can store a link persistently by adding it to the personal iServer instance.

While the quality of the persistent link data can be ensured by controlling the users who have access to a personal iServer instance, we do not have any direct control mechanism to guarantee the integrity of information provided by remote iServer peers. In a meeting setting this integrity might not be a problem but we will also discuss solutions for more open settings in Sect. 6.

4 P2P iServer Architecture

The general interaction between peers consists of sending single API calls from one peer to another, the execution of that request on the remote site and the transfer of the result back to the initiating peer. The information returned by the remote peers has to be combined and integrated with information that is available from the local iServer instance. The functionality of a remote iServer system is offered to a local iServer instance via a peer service. This service is implemented within a peer service platform that separates the aspects of *interaction* and *connection*.

The interaction architecture is shown in Fig. 3. For the sake of simplicity, we reduce service interaction to the transfer of request and response data between a requesting local system shown on the left-hand side and a responding remote system on the right-hand side. One-to-many and many-to-one interactions which typically occur in distributed iServer scenarios can always be represented by multiple one-to-one data transfers. We now describe the main steps of interaction labeled ⓐ to ⓗ in the figure.

ⓐ In the local system, a request handler is used to post a local request. Depending on the concrete peer service, data may have to be provided by the requesting client. ⓑ The handler creates a message object containing the request data and sends its XML string representation to a remote request handler. ⓒ On the remote site, the message is reconstructed from the string value received and

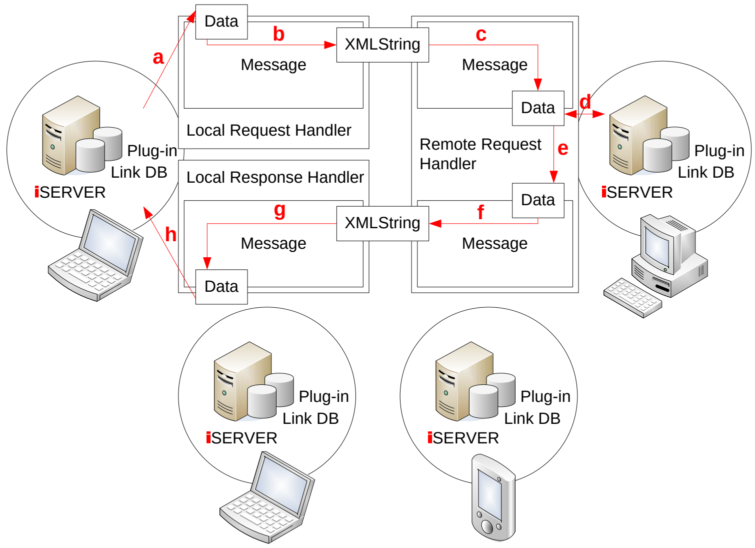

Fig. 3. Interaction architecture for peer services

the request data is extracted. ⓓ The remote request handler then processes the request which includes access to the iServer API yielding a response data object. ⓔ This response data object is wrapped with a message object and ⓕ its XML string representation is sent back to the local response handler. ⓖ The response handler reconstructs the message and extracts the response data. ⓗ Typically, the response is then integrated into the local iServer link structure.

We can identify three components each responsible for a particular aspect of service interaction: *handlers*, *data* and *message* objects. A `Data` class encapsulates the iServer API call while the `Message` class enables content-independent implementation of message formats such as plain XML, compressed or encrypted representations. Handlers implement the data object processing which consists of executing API calls and integrating results.

A request handler relies on a connection service providing the means to address a particular or multiple peers and to send and receive messages. This functionality is encapsulated in a connection service component. It consists of a peer abstraction that can be identified by other peers, a class representing the group of peers sharing links as well as resources and a connection handler creating and maintaining physical channels to remote peers.

5 Application

We now explain how applications built on top of the P2P iServer architecture can support the kind of scenario described in Sect. 2. The iServer API allows resources to be created as well as links to be defined between source and target resources. The set of resources and links therefore form a graph where nodes are resources and directed edges represent links. Each iServer instance manages a graph containing the resources and links stored locally and is able to transiently integrate links received from other iServer instances. A general authoring tool allows resources to be added by the user and new links to be created.

Using this authoring tool, a user may, for example, select a single slide of a particular slideshow as a source and link it to parts of a video resource. For this purpose, a PowerPoint iServer plug-in could be used that allows PowerPoint slideshows to be integrated as resources and provides the means to use each slide as a selector. Furthermore, an iServer movie plug-in introduces a movie resource type as well as a selector for using particular frames or sequences to be used as link sources or targets. Other resource types can easily be integrated into applications by developing new iServer plug-ins.

In our meeting scenario, assume a participant created a link from a presentation slide to a video before the meeting and that both the slideshow and video are stored on the company server and have been replicated on the user's laptop computer. The link would be shared with the company's iServer instance whenever the user entered the meeting room and was detected as a peer. As a result, during the meeting, they are able to show the slide on a wall projection and activate the link to the video.

Further, assume an interactive paper version of a brochure has been edited by a user on the way to the meeting. Using a digital pen, all annotations made on the brochure have been digitally captured and stored as linked resources in the user's iServer instance. The iPaper plug-in allows such annotations to be linked from particular areas of the brochure such as words, paragraphs or pictures. After entering the meeting room, the annotations would be shared with the company's iServer and the interactive table could be used to show the digital brochure document enriched with that user's annotations without them having to explicitly copy any information.

6 Discussion

The main interaction of iServer peers consists of exchanging resources and links. A common request is a query for in- and outgoing links from a particular resource as well as the linked resources. In the meeting setting, it can safely be assumed that users share data with trustworthy users and that the amount of data received is manageable. However, in other settings such as an educational environment where students may exchange annotations and additional information related to lectures, this is no longer the case. As a result, the user receives a potentially large set of links and resources from possibly unknown users.

Our framework supports individual users in deciding on their own levels of trust in certain users. A user who knows and is therefore able to rate a relatively small number of other users can infer ratings from a vast number of other users by exploiting the transitive propagation of trust.

Since a user receives responses from multiple users, the information on which a filtering decision is based consists of the sending user and the currently received item as well as the set of previously received items. In order to exploit all information available, we also take into account the frequency with which a particular item has been received. Therefore, we combine user ratings with response ratings to filter responses returned from remote peers. This may, not only help to improve the quality of information presented to the user, but also reduce the quantity, thereby preventing the information overload that could result from a large, highly-connected information space.

7 Conclusions

We have presented a notion of cooperative information spaces based on community-based authoring of links between arbitrary resources and elements within these resources. While many of the features of the iServer framework and its cooperative version can be found in other systems developed within the hypertext community, the main contribution of our work is to combine these in a single, extensible framework. Further, in contrast to most other approaches, we do not use the P2P functionality to distribute data persistently but rather to build a supplemental transient link layer on top of a user's personal information space.

References

1. D. DeRoure, L. Carr, W. Hall, and G. Hill. A Distributed Hypermedia Link Service. In *Proc. of SDNE '96, 3rd Workshop on Services in Distributed and Networked Environments*, Washington, DC, USA, 1996.
2. K. Grønbæk, J. Kristensen, P. Ørbæk, and M. A. Eriksen. Physical Hypermedia: Organising Collections of Mixed Physical and Digital Material. In *Proc. of Hypertext 2003, 14th Conference on Hypertext and Hypermedia*, Nottingham, UK, August 2003.
3. J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, and R. R. Swick. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proc. of WWW10, 10th Intl. World Wide Web Conference*, Hong Kong, May 2001.
4. R. D. Larsen and N. O. Bouvin. HyperPeer: Searching for Resemblance in a P2P Network. In *Proc. of Hypertext 2004, 15th ACM Conference on Hypertext and Hypermedia*, Santa Cruz, USA, August 2004.
5. D. T. Michaelides, D. E. Millard, M. J. Weal, and D. De Roure. Auld Leaky: A Contextual Open Hypermedia Link Server. In *Proc. of OHS-7, 7th Intl. Workshop on Open Hypermedia Systems*, August 2001.
6. D. Millard, L. Moreau, H. Davis, and S. Reich. FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability between Hypertext Domains. In *Proc. of Hypertext 2000, Conference on Hypertext and Hypermedia*, May 2000.
7. Beat Signer, Michael Grossniklaus, and Moira C. Norrie. Interactive Paper as a Mobile Client for a Multi-Channel Web Information System. *World Wide Web Journal*, 10(4), December 2007.
8. Beat Signer and Moira C. Norrie. As We May Link: A General Metamodel for Hypermedia Systems. In *Proc. of ER 2007, 26th Intl. Conference on Conceptual Modeling*, Auckland, New Zealand, November 2007.
9. Axel Vogelsang and Beat Signer. The Lost Cosmonaut: An Interactive Narrative Environment on Basis of Digitally Enhanced Paper. In *Proc. of VS 2005, 3rd Intl. Conference on Virtual Storytelling*, Strasbourg, France, December 2005.